UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/088,988	07/31/2002	Xiaoning Nie	1406/52	9022

25297 7590 12/14/2007

JENKINS, WILSON, TAYLOR & HUNT, P. A.
3100 TOWER BLVD., Suite 1200
DURHAM, NC 27707

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	PAPER NUMBER
2183	
MAIL DATE	DELIVERY MODE
12/14/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/088,988	NIE, XIAONING	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 October 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 July 2007 & 12 October 2007* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1 and 3 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment and Affidavit(s) as received on 10/12/2007.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kadosumi et al., U.S. Patent No. 5,870,620 (as applied in the previous Office Action and herein referred to as Kadosumi) in view of Intel, "IA-64 Application Developer's Guide," 1999 (herein referred to as Intel).

6. Referring to claim 1, Kadosumi has taught a method for processing conditional jump instructions in a processor with pipeline computer architecture, the method comprising:

a) loading and decoding a processor instruction, the processor instruction containing an instruction opcode and a post-condition, which specifies that a conditional jump is to be processed. See Fig.9, and note the ADD_SW2 instruction. The examiner deems the loading and decoding of this instruction to be inherent as each instruction must be loaded and decoded before execution. Furthermore, each instruction inherently includes an opcode because the opcode specifies the operation to be performed. For instance, ADD_SW2 has a bit pattern in its encoding which specifies to the system that it is an ADD_SW2 instruction. Finally, the instruction includes a post-condition as claimed because, when this instruction bit pattern is encountered, it tells the system that a conditional jump is to be processed, as seen in Fig.9 and column 13, lines 48-57.

b) Kadosumi has not explicitly taught that the processor instruction contains register addresses, and a relative jump distance. However, Official Notice is taken that it is well known and accepted in the art that ADD instructions specify two registers to be added and that branch instructions include relative jump distances. Clearly, if one is adding two numbers, which is the case in Fig.9 with the ADD_SW2 instruction, two operands must be specified. A third register specified could be a destination register. By specifying register values instead of immediate values, one could save on the amount of bits used to encode the instruction. That is, instead of encoding the entire value to be added, one merely needs to encode the register that the value is held in, and there are much fewer registers than possible values. Furthermore, since the ADD_SW2 is also a conditional branch instruction, a jump address must be specified. Relative addresses are known in the art as being advantageous because the entire jump address does not need to be encoded in the instruction. Instead, only the offset needs to be encoded and

eventually added to the program counter, and the offset requires much fewer bits. As a result, in order to achieve the disclosed functionality of the ADD_SW2 instruction and to minimize the number of bits required to encode the instruction, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kadosumi such that the processor instruction includes register addresses and a relative jump address.

c) Kadosumi has not taught a precondition, which comprises at least one precondition bit that specifies under which conditions the instruction is actually to be executed. However, Intel has taught the use of a predicates as preconditions. See pages 3-4 and 4-7. That is, all but a select few instructions in Intel's architecture contain a precondition (a predicate), which allows for conditional execution of that instruction. When the predicate is set, the instruction will execute, and when it's not set, the instruction will not execute. See page 7-3 and note that ADD instructions are even predicated. A predicate allows for greater control and flexibility of instruction execution. All instructions can be conditionally executed depending on the state of the system. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kadosumi to include a precondition as taught by Intel.

d) Kadosumi has not explicitly taught that the post-condition specifies that the corresponding flag bits of an arithmetic-logic unit are to be checked, wherein the post-condition comprises at least one post-condition bit that is checked in the processor. However, first note that the ADD instruction of Fig.9 produces a result whose value must be checked in order to determine branching. Specifically, the result is checked to see if it is zero or more, or less than zero. Official Notice is taken that arithmetic status flags are well known and accepted in the art. Such flags are set after every arithmetic operation and indicate overflow, zero, negative, interrupt, and

sometimes more statuses. So, in order to check if a result is zero or more, or less than zero, this can be done a number of ways with arithmetic flags. The system can first check the Z (zero) flag, and if set, then the system would branch or fall through to code block ND2 (in Fig.9) because the result is zero. If the Z flag is not set, then the system could check the N (negative flag) to determine if the result is negative. If set, then the system would branch or fall through to code block ND3 because the number is negative; otherwise it would branch or fall through to code block ND2 because it is positive. As a result, because the system needs to check the value of the result in order to determine branching in Fig.9, and arithmetic flags facilitate such checking, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Kadosumi such that the post-condition specifies that the corresponding flag bits of an arithmetic-logic unit are to be checked, wherein the post-condition comprises a plurality of post-condition bits that are checked in the processor.

e) Kadosumi, as modified by Intel, has further taught checking the precondition, and executing the decoded processor instruction if the precondition is fulfilled. Again, recall from Intel that predicates allow for conditional execution of instructions. So, if the predicate associated with ADD_SW2 (of Kadosumi) is set (i.e., the precondition is fulfilled), then the instruction will be executed.

f) Kadosumi, as modified by Intel, has further taught that in the case of a fulfilled precondition, checking the post-condition, and carrying out no jump if the post-condition is not fulfilled, and checking the corresponding flag bits, if the post-condition is fulfilled. See Fig.9(a) of Kadosumi and note that after the precondition is fulfilled (i.e., after it is determined that the ADD_SW2 predicate is set and, consequently, that the ADD_SW2 instruction is to actually execute), a post-

condition is checked. Clearly, not all instructions require the checking of arithmetic flags (for instance, a NOP or an increment instruction would not require checking flags), and therefore the bit pattern of the ADD_SW2 instruction acts as a post-condition which results in the system processing a conditional branch and checking flags to perform said branch. This bit pattern is the post-condition.

g) Kadosumi, as modified by Intel, has further taught jumping to a jump address as a function of the relative jump distance contained in the processor instruction if the post-condition is fulfilled and the checked flag bits are set. If the flags are set in the appropriate manner, then jumping to the appropriate code block will occur in Kadosumi, Fig.9. Also, recall that it would have been obvious to include a relative jump address in the instruction because it requires less bits and can be added to the program counter to obtain the full jump address.

7. Referring to claim 3, the apparatus of claim 3 performs the method of claim 1. Consequently, claim 3 is rejected for the same reasons set forth in the rejection of claim 1.

Response to Arguments

8. Applicant's arguments filed on October 12, 2007, have been fully considered but they are not persuasive.

9. Applicant argues the novelty/rejection of claims 1 and 3 on pages 10-11 of the remarks, in substance that:

> "The Intel Guide has a date of copyright of 1999. Further, on its face, the Intel Guide has a date of May 1999. However, it is not clear if this May 1999 date is an actual date of publication where it was actually distributed to anyone in the public... the printed date on Intel's "IA-64 Application Developer's Architecture Guide", i.e. May 1999, is very likely not the date this document was disclosed to the public, if at all. It is rather likely that Inters "IA-64 Application Developer's Architecture Guide" was made to the public, if at all, around the time the Itanium processor was actually available, i.e. in 2001."

10. These arguments are not found persuasive for the following reasons:

a) The examiner asserts that none of the speculation on applicant's behalf proves that the Intel reference was not published and made available to the public in May 1999. First, even if the Itanium wasn't released until 2001, this doesn't imply that IA-64 wasn't released to the public earlier. In fact, the Wikipedia article found at http://en.wikipedia.org/wiki/64-bit, and attached herein, includes a timeline showing that Intel released the IA-64 instruction set to the public in 1999, two years prior to the release of Itanium. Furthermore, a quick Google search for the Intel reference yields a citeseer link which lists published documents that cite the specific Intel reference. At least one of these documents was published in 1999 by Rutgers University employees. See the provided reference "Instruction Scheduling in the presence of Java's Runtime Exceptions", which is cited as extrinsic evidence for supporting the examiner's position. This document was published in 1999 and it includes a citation to the Intel reference ([10]). Consequently, the guide must've been made available to the university employees, i.e., the public, well before 2001. Even further, the must've been made available to the university employees prior to August 1999 (the publication date). Therefore, the evidence points to this document being published well before applicant's argued date. Consequently, the examiner asserts that the reference and rejection are valid.

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

DJH
David J. Huisman
November 8, 2007